# Smart Light

[Smart Light 1](#_Toc519168773)

[Pictures 2](#_Toc519168774)

[I. Introduction 3](#_Toc519168775)

[II. Smart Light 3](#_Toc519168776)

[1. Hardware 3](#_Toc519168777)

[a) Esp8266 Nodecmu 3](#_Toc519168778)

[b) Avoidance Module 4](#_Toc519168779)

[c) Photoresistor Module 4](#_Toc519168780)

[d) RGB LED Module 5](#_Toc519168781)

[2. Software 6](#_Toc519168782)

[a) Arduino 6](#_Toc519168783)

[b) ESP8266 FreeRTOS 8](#_Toc519168784)

[c) Web Design 10](#_Toc519168785)

[3. Operation of Smart Light 11](#_Toc519168786)

[4. Advantages and disadvantages 13](#_Toc519168787)

[a) Advantages 13](#_Toc519168788)

[b) Disadvantages 13](#_Toc519168789)

[5. Future development 13](#_Toc519168790)

# Pictures

[Figure 1. Esp8266 Nodecmu 3](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158909)

[Figure 2. Avoidance Module 4](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158910)

[Figure 3. Photoresistor Module 5](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158911)

[Figure 4. RGB LED Module 5](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158912)

[Figure 5. Web Server 10](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158913)

[Figure 6. Working principle IR 11](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158914)

[Figure 7. smart light diagram 12](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158915)

[Figure 8. Operation 12](file:///C:\Users\Administrator\Desktop\Smart_Light_eng.docx#_Toc519158916)

# Introduction

Update the trend of Smart Light development in Appliances and Industry, not only so Smart Light also applied in Smart Car Technology such as identified light, identifying gesture people... Smart Light is applied in many fields; this project is the choice for newbies to learning about embedded programming... In this project, our team introduces about Smart Light Application in identified daytime light and gesture of people to turn on/ turn off the light, detect impediment .

# Smart Light

Hardware

### Esp8266 Nodecmu

After many years of development, there are now more than 14 ESP versions have been released, of which the most popular are the ESP-12. The ESP-12 module incorporates the ESP8266 firmware on the Arduino and the standard communication hardware design that makes up the NodeMCU, kit Development ESP8266 common in the current. With usage, connect easy, you can programming, load the live program on the Arduino software, same as interactive with the Arduino library available, NodeMCU is the first choose for you want to find understand the current ESP8266.



Figure 1. Esp8266 Nodecmu

### Avoidance Module

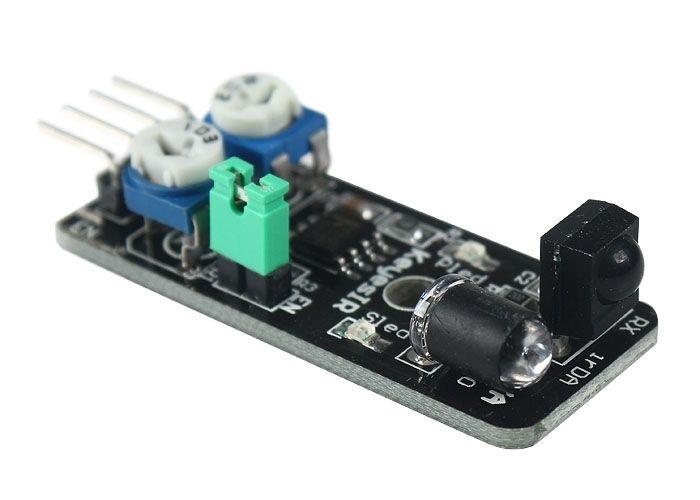
IR-reflection sensor, useful for obstacle avoidance applications. When an obstacle is in front of the IR sender/receiver the ‘Out’ pin is switched low (active low). Working voltage 3.3V-5V.The circuit sensitivity can be adjusted with a pot. The obstacle detection distance can be adjusted up to approximately 7cm. An enable (EN) jumper can be fitted for continuous operation. Removal of the EN jumper allows an external logic signal (at the EN pin) to switch the detector on and off (low = active, high = off).

Figure 2. Avoidance Module

### Photoresistor Module

Photoresistor Module also is known as Light Dependent Resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices. LDR used to measure light intensity. It can determine the presence or absence of light.  
This module consists of a photoresistor and a 10 kΩ in-line resistor:  
Turn left (look in the direction from bottom to optical): you will increase the sensitivity of the sensor with light: only need a small amount of light will break the circuit.  
Turn to the right: You will reduce the sensitivity of the sensor to light, requiring more intense light to switch off the circuit. The photoresistor's resistance will decrease in the presence of light and an increase in the absence of it. The output is analog and determines the intensity of light (Operating Voltage: 3.3V to 5V & Output type: Analog).  
Input: middle (+) +5V  
(-) GND  
Output: (S) signal 1/0 (True/False).

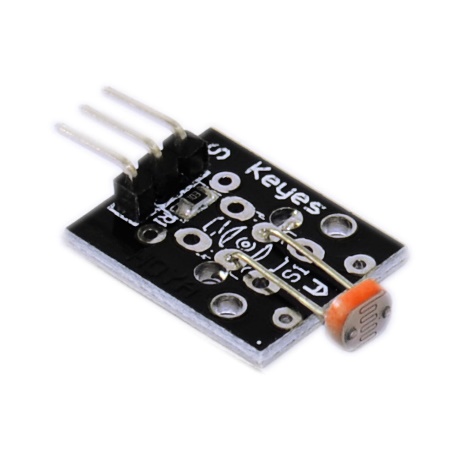


Figure 3. Photoresistor Module

### RGB LED Module

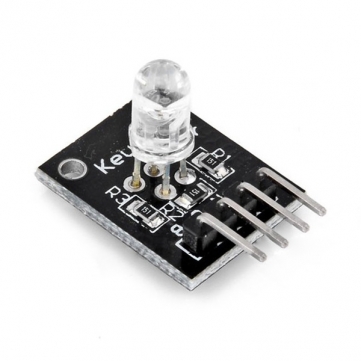
An RGB LED is a combination of 3 LEDs in just one package:  
1x Red LED  
1x Green LED  
1x Blue LED  
The 3 LEDs can share the cathode, this results in an RGB LED that has 4 pins, one for each LED, and one common cathode. You can create one of those three colors – red, green or blue – by activating just one LED, or you can produce other colors, you can combine the three colors in different intensities. To generate different colors you can use PWM to adjust the brightness of each LED.  
Input: R: +5V (The anode of Red LED)  
B: +5V (The anode of Blue LED)  
G: +5V (The anode of Green LED)  
GND: GND  
Output: The light

Figure 4. RGB LED Module

## Software

### Arduino

Arduino's integrated development environment(IDE) is a cross platform application written in Java , and from this IDE it will be used for the programming language and for the project. It is designed for beginners familiar with the field of software development. Users only need to define two functions to create a program loop execution (cyclic executive) can run:

* setup()This function runs every time a program is started, used to set the settings
* loop()This function is called repeated until the circuit board is shut off

**Code smart light**

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

const char\* ssid = "Hacker";

const char\* password = "12345678";

int sound\_sensor = D1;

int red = D5;

int green = D6;

int blue = D7;

int clap = 0;

bool isClap = false;

long start = 0;

long range = 0;

bool rLight = LOW;

bool gLight = LOW;

bool bLight = LOW;

void setup() {

// put your setup code here, to run once:

pinMode(sound\_sensor, INPUT);

pinMode(red, OUTPUT);

pinMode(green, OUTPUT);

pinMode(blue, OUTPUT);

Serial.begin(115200);

digitalWrite(red, LOW);

digitalWrite(green, LOW);

digitalWrite(blue, LOW);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("Connected");

}

void loop() {

int status\_sensor = digitalRead(sound\_sensor);

if (status\_sensor == 0) {

if (clap == 0) {

start = millis();

Serial.println(start);

}

if (clap == 1) {

range = millis();

Serial.println(range);

}

isClap = true;

}

else {

if (isClap) {

clap++;

Serial.println(clap);

isClap = false;

}

}

if (clap == 1 && millis() - start > 1000) {

digitalWrite(red, !rLight);

rLight = !rLight;

clap = 0;

}

if (clap == 2 && millis() - range > 1000) {

digitalWrite(green, !gLight);

gLight = !gLight;

clap = 0;

}

if (clap == 3) {

digitalWrite(blue, !bLight);

bLight = !bLight;

clap = 0;

}

if (WiFi.status() == WL\_CONNECTED) {

HTTPClient http;

String url = "http://smartlight.ddns.net/control.php";

http.begin(url);

http.GET();

String control = http.getString();

http.end();

if (control == "on1") {

digitalWrite(red, HIGH);

rLight = HIGH;

clap = 0;

}

else if (control == "off1") {

digitalWrite(red, LOW);

rLight = LOW;

clap = 0;

}

if (control == "on2") {

digitalWrite(green, HIGH);

gLight = HIGH;

clap = 0;

}

else if (control == "off2") {

digitalWrite(green, LOW);

gLight = LOW;

clap = 0;

}

if (control == "on3") {

digitalWrite(blue, HIGH);

bLight = HIGH;

clap = 0;

}

else if (control == "off3") {

digitalWrite(blue, LOW);

bLight = LOW;

clap = 0;

}

}

delay(15);

}

### ESP8266 FreeRTOS

Developed in partnership with the world's leading chip companies over a 14 year period, the FreeRTOS Kernel is a market leading real time operating system (or [RTOS](https://esp8266.vn/freertos-sdk/freertos-sdk/)), and the de-facto standard solution for microcontrollers and small microprocessors.

FreeRTOS is truly free and supported, even when used in commercial applications. The FreeRTOS open source MIT lincense does not require you to expose your proprietary IP. You can take a product to market using FreeRTOS without even talking to us, let alone paying any fees, and thousands of people do just that

**Code smart light**

#include "esp\_common.h"

#include "freertos/FreeRTOS.h"

#include "freertos/task.h"

#include "uart.h"

#include "gpio.h"

static int led\_state = 1;

#define LED\_GPIO14 14

#define LED\_GPIO12 12

#define LED\_GPIO13 13

#define LOW 0

#define HIGH 1

int mode = 1;

void ICACHE\_FLASH\_ATTR task\_led\_dark\_light()

{

uint16 temp;

for(;;){

vTaskDelay(100);

temp = system\_adc\_read(); //read analog from sensor

if (temp <= 700) //sensor return <=700 => light

{ //turn off light

GPIO\_OUTPUT\_SET(LED\_GPIO14, LOW);

GPIO\_OUTPUT\_SET(LED\_GPIO12, LOW);

GPIO\_OUTPUT\_SET(LED\_GPIO13, LOW);

//printf("%d",temp,"\n");

}

if (temp >= 800) //sensor return >= 800 => dark

{ //turn on Light

GPIO\_OUTPUT\_SET(LED\_GPIO14, LOW);

GPIO\_OUTPUT\_SET(LED\_GPIO12, HIGH);

GPIO\_OUTPUT\_SET(LED\_GPIO13, LOW);

//printf("%d",temp,"\n");

}

}

}

void change\_mode(){

if(mode == 0) {

mode = 1;

} else {

mode = 0;

}

}

}

bool rLight = LOW;

bool gLight = LOW;

bool bLight = LOW;

void ICACHE\_FLASH\_ATTR task\_blink\_red(){

GPIO\_OUTPUT\_SET(LED\_GPIO14, !rLight);

rLight ^= 1;

}

void ICACHE\_FLASH\_ATTR task\_blink\_green(){

GPIO\_OUTPUT\_SET(LED\_GPIO12, !gLight);

gLight ^= 1;

}

void ICACHE\_FLASH\_ATTR task\_blink\_blue(){

GPIO\_OUTPUT\_SET(LED\_GPIO13, !bLight);

bLight ^= 1;

}

void ICACHE\_FLASH\_ATTR task\_off(){

GPIO\_OUTPUT\_SET(LED\_GPIO14, LOW);

rLight = LOW;

GPIO\_OUTPUT\_SET(LED\_GPIO12, LOW);

gLight = LOW;

GPIO\_OUTPUT\_SET(LED\_GPIO13, LOW);

bLight = LOW;

}

//task for avoidance sensor

void ICACHE\_FLASH\_ATTR task\_led\_impediment()

{

int i = 0;

for (;;){

int pre = 0;

int count = 0;

if (!GPIO\_INPUT\_GET(5)){

pre = 1;

count = 1;

for (i = 0;i < 200;i++){

if (!pre){

if (!GPIO\_INPUT\_GET(5)){

count++;

pre = 1;

}

}else{

if (GPIO\_INPUT\_GET(5)){

pre = 0;

}

}

os\_delay\_us(5000);

}

printf("%d\n", count);

switch(count){

case 1:

task\_blink\_red();

break;

case 2:

task\_blink\_green();

break;

case 3:

task\_blink\_blue();

break;

default:

change\_mode();

break;

}

}

os\_delay\_us(1000);

}

}

void user\_init(void)

{

// Change baudrate UART0 to 115200 for printf()

UART\_ConfigTypeDef uart\_config;

uart\_config.baud\_rate = BIT\_RATE\_115200;

uart\_config.data\_bits = UART\_WordLength\_8b;

uart\_config.parity = USART\_Parity\_None;

uart\_config.stop\_bits = USART\_StopBits\_1;

uart\_config.flow\_ctrl = USART\_HardwareFlowControl\_None;

uart\_config.UART\_RxFlowThresh = 120;

uart\_config.UART\_InverseMask = UART\_None\_Inverse;

UART\_ParamConfig(UART0, &uart\_config);

PIN\_FUNC\_SELECT(GPIO\_PIN\_REG\_12, FUNC\_GPIO12);

PIN\_FUNC\_SELECT(GPIO\_PIN\_REG\_13, FUNC\_GPIO13);

PIN\_FUNC\_SELECT(GPIO\_PIN\_REG\_14, FUNC\_GPIO14);

PIN\_FUNC\_SELECT(GPIO\_PIN\_REG\_5, FUNC\_GPIO5);

xTaskCreate(task\_led\_dark\_light, "task\_led\_dark\_light", 256, NULL, 2, NULL);

xTaskCreate(task\_led\_impediment, "task\_led\_impediment",256, NULL, 2 , NULL);

}

### Web Design

Web design control turn on, turn off from the server

Web design in HTML, CSS, JavaScript, PHP

Displaying light led from NodeCMU on host

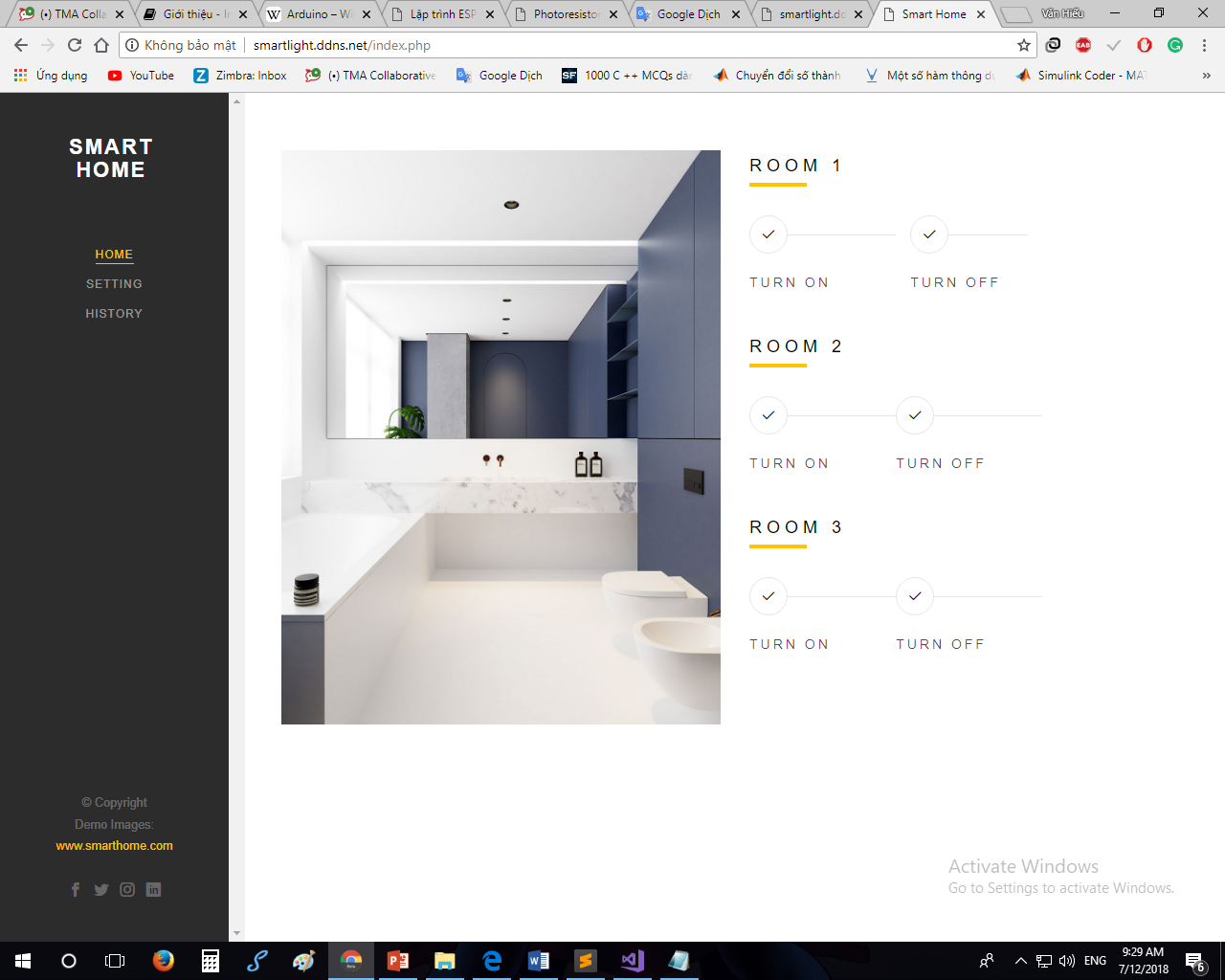
[](http://smartlight.ddns.net)

Figure 5. Web Server

## Operation of Smart Light



RGB led light when responding enough the sensor condition

**Avoidance Moudule**

The IR transmitter sends an infrared signal that, in case of a reflecting surface (e.g. white color), bounces off in some directions including that of the IR receiver that captures the signal detecting the object.

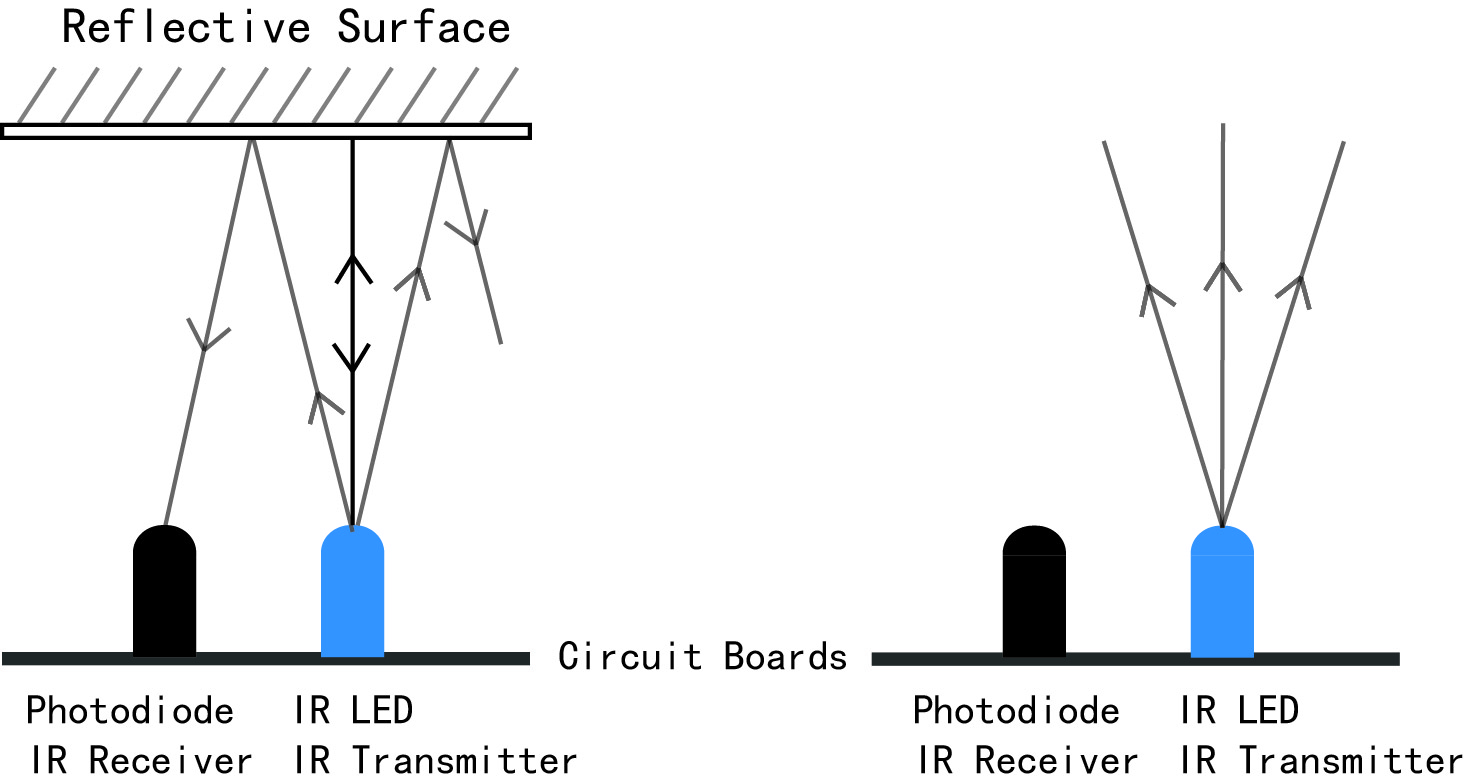


Figure 6. Working principle IR

When the surface is absorbent (e.g. black color) the IR signal isn’t reflected and the object cannot be detected by the sensor. This result would occur even if the object is absent.

**Photoresistor Module**

When the value read from the photo resistor sensor module goes below the threshold value, i.e. it becomes dark, the RGB LED is switched on. The LED is switched off when the analog value from the sensor goes above the threshold value.

A threshold value can be selected by using the previous sketch to determine the analog value at the desired light level when the LED should be triggered. The value assigned to the threshold variable at the top of the sketch can then be set to the new value

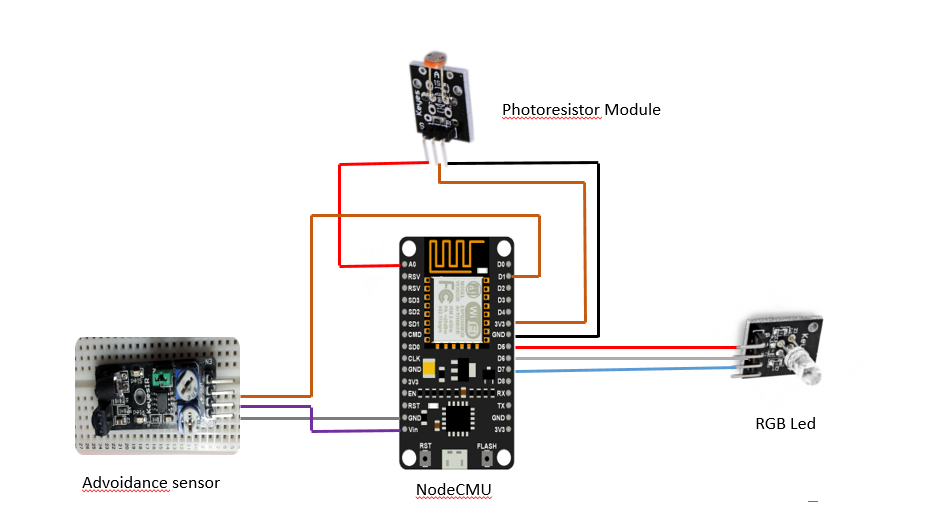


Figure 7. smart light diagram

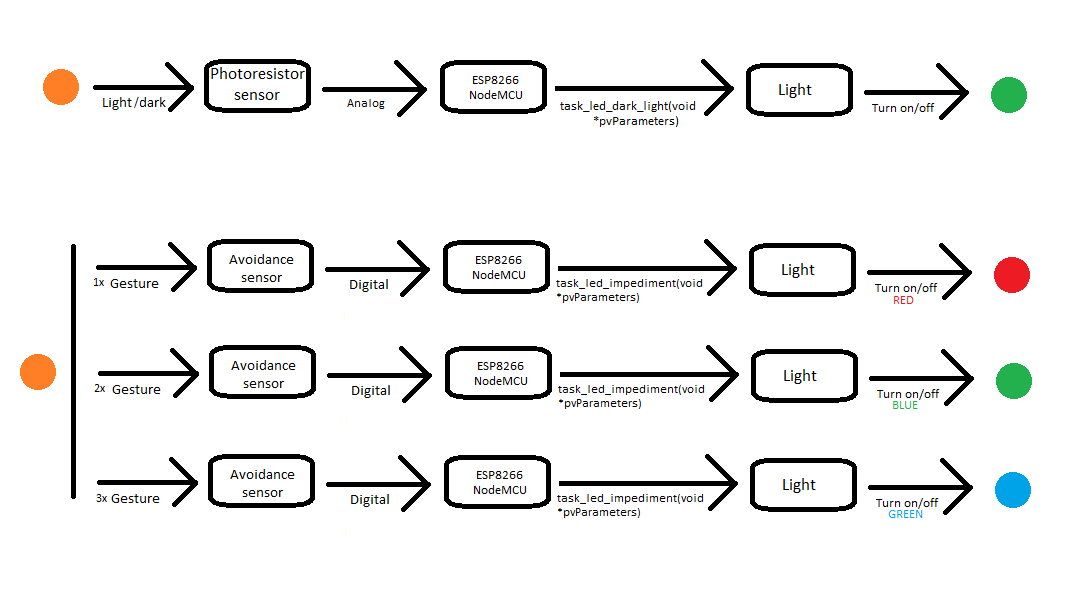


Figure 8. Operation

## Advantages and disadvantages

### Advantages

* The system is compact, easy to install
* The photoresistor module has the Infrared Obstacle Avoidance Sensor has Power, Ground, Signal, and Enable Pin
* Can develop more applications than the observation...

### Disadvantages

* Sensor is weak so sensitivity is low and delayed
* Delay up to a few seconds

## Future development

* Retreat
* Detecting light
* Predictable collisions (front and behind)